 DOI: 10.3217/978-3-85125-663-5-40

Detecting Out-of-Distribution Traffic Signs

Madhav Iyengar[1], Michael Opitz[2] and Horst Bischof[2]

Abstract— **This work addresses the problem of novel traffic sign detection, i.e. detecting new traffic sign classes during test-time, which were not seen by the classifier during training. This problem is especially relevant for the development of autonomous vehicles, as these vehicles operate in an open-ended environment. Due to which, the vehicle will always come across a traffic sign that it has never seen before. These new traffic signs need to be immediately identified so that they can be used later for re-training the vehicle. However, detecting these novel traffic signs becomes an extremely difficult task, as there is no mechanism to identify from the output of the classifier whether it has seen a given test sample before or not. To address this issue, we pose the novel traffic-sign detection problem as an out-of-distribution (OOD) detection problem. We apply several state-of-the-art OOD detection methods and novelty detection methods on the novel traffic-sign detection problem and also establish a benchmark using the German Traffic Sign Recognition Benchmark dataset(GTSRB). In our evaluation, we show that both out-of-distribution approaches and novelty detection approaches are suitable for OOD traffic sign detection.**

I. Introduction

The problem of detecting novel traffic signs is to detect whether a given traffic sign is from a class which was not seen by the classifier during training. This is crucial for autonomous vehicles, as they navigate within an open-ended environment and therefore, come across previously unseen traffic signs regularly. Consequently, these novel traffic signs have to be recognized, so that they can be labeled and added to the ever-increasing training data of the vehicle. However, the collection of this data is an extremely difficult task, since there is no way to ascertain from the output of the classifier whether a given test sample is similar to the training data or not. Even though modern neural networks manage to attain state-of-the-art performance in several complex tasks like image classification. [12], medical image diagnosis [1], speech recognition [10], natural language processing [19], etc., they are typically overconfident in their predictions. Several recent works substantiate this and show that neural networks give high predictions even on irrelevant, [11], [26], [20] and unrecognizable, [21] inputs. To overcome this problem of collecting a vast amount of traffic sign data, we propose to present the novel traffic sign detection problem as an out-of-distribution (OOD)/ novelty detection problem.

[1]This work was performed during an internship at the Institute of Computer Graphics and Vision, {thealmightylion.madhav}@gmail.com

[2]Institute of Computer Graphics and Vision, Graz University of Technology {michael.opitz,bischof}@icg.tugraz.at

The main objective of OOD detection is to detect whether a given test sample is from the in-distribution (i.e. same distribution as the data on which the network was trained on) or from the out-distribution (i.e. a distribution different from the in-distribution). This can also be interpreted as an additional binary classification task, where we want to predict true, if the given data is from the in-distribution and false, if it is not. However, it is crucial to add this binary classification task without affecting the performance of the original classifier.

A naïve solution to the OOD problem is to increase the size of the training data and explicitly add OOD examples to it. This enables us to teach the network to classify whether a test sample is in-distribution or OOD, by just using an additional label. However, collecting such a dataset is prohibitively expensive, as OOD samples, by definition, can be infinitely many. Furthermore, with the addition of these OOD samples more complex neural network architectures may need to be employed to correctly classify the training samples. This makes training of the network intractable and eventually makes this approach computationally expensive.

Thus, to solve this challenging real-world problem of novel traffic-sign detection, we adopt state-of-the-art OOD detection methods, i.e. [16], [14], and also apply non deep-learning based novelty detection methods like the One-Class Support Vector Machine (SVM) [23]. Further, we also evaluate a supervised linear SVM to this problem, in order to get an estimate of the upper bound accuracy of the methods. and linear SVM [3]. Except the supervised Linear SVM, none of these methods need OOD samples during training time. We illustrate how we use OOD detectors to ease the labeling task of traffic signs in Figure 1. To the best of our knowledge, we are the first to establish a benchmark on the detection of OOD traffic signs. We accomplish this task by using the German Traffic Sign Recognition Benchmark (GTSRB) [25] and also test the performance of our trained classifiers on a private dataset with extremely promising results. These methods manage to achieve a high AUROC score of 97.2% and a very low detection error of 5.7% on the challenging task of detecting novel classes in the GTSRB dataset.

II. Related Work

Out-of-distribution detection has received a lot of attention recently. Approaches can be mainly categorized into simple threshold based detectors, GAN based approaches and works that directly estimate confidence.

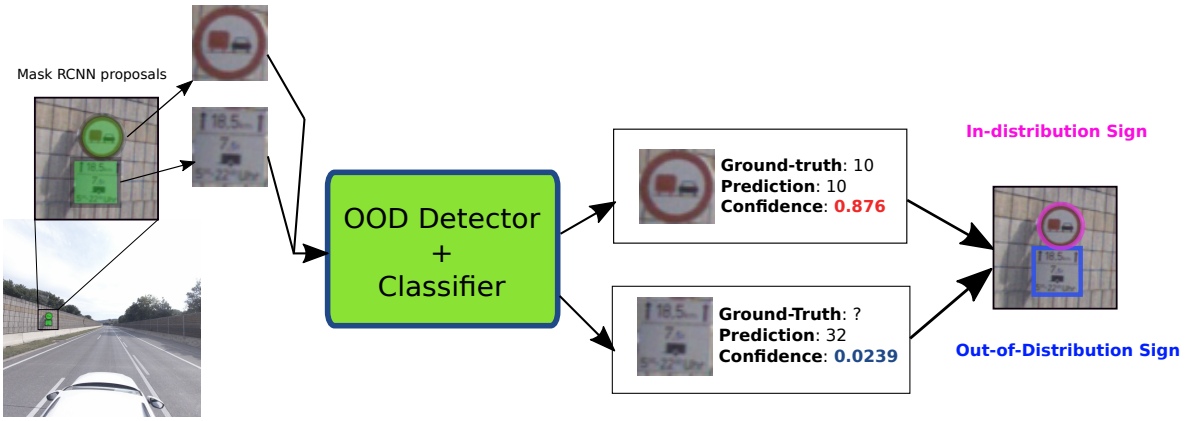

Fig. 1: Pipeline to ease labeling task of traffic-signs. The first step of our pipeline is to detect traffic-signs using Mask RCNN. The second step is to extract the traffic-sign crops from the predicted proposals. In the third step, our OOD detector provides a confidence score for each input sample. If the confidence score is greater than a selected threshold, i.e. in-distribution, the input sign does not require labeling. Whereas, if the confidence score is less than the threshold, i.e. OOD, the input sign should be marked for labeling.

Threshold Based Detectors : Hendrycks and Gimpel [11] propose a baseline method to detect OOD samples by using a simple threshold based detector mechanism, which requires no additional re-training of a network. Their method is based on the empirical observation that well-trained neural networks tend to assign higher softmax scores to samples which are from the in-distribution compared to samples from OOD. They use the predictions of a pre-trained classifier to compute a confidence score $c(x)$ on a test sample x. The detector then assigns the label 1 if the confidence score $c(x)$ is greater than some chosen threshold δ and 0 otherwise. This confidence score c is defined as the maximum value of the predictive distribution of the neural network.
Liang et al. [16] propose ODIN, which builds upon this confidence threshold based method and further enhances its performance by applying temperature scaling [9] and by adding small controlled perturbations [8] to the input data. Liang et al. [16] use these controlled perturbations to increase the softmax score assigned by the network on a given input. They show that adding these perturbations to the input combined with temperature scaling, helps in further enlarging the gap between the softmax scores assigned to in-distribution and OOD examples.
These techniques, although computationally cheap, heavily rely on the ability of the pre-trained classifier to separate the softmax scores on in-distribution and OOD samples.

Confidence Calibration using GAN : Lee et al. [14] propose a training mechanism which focuses on minimizing the Kullback-Leibler (KL) divergence loss between the softmax scores assigned by the network on OOD samples and the uniform distribution. Minimizing this loss forces the network to be uncertain (i.e. unable to assign a high softmax score) on examples which are not from the in-distribution. However, to minimize the KL divergence, OOD samples are required. To obtain the training data for this loss, the authors propose introducing a Generative Adversarial Network (GAN) [7] based loss which will be responsible to generate the OOD samples. However, unlike the original GAN, they modify the GAN loss so that it generates samples which are in the the low density region of the in-distribution. These generated samples are used as the OOD dataset to minimize the KL divergence. The model is then trained jointly with this KL divergence loss, GAN loss and the original classification loss.

Confidence Estimation : DeVries and Taylor [5] propose a training method which directly outputs the confidence of a network on a given sample. They achieve this by introducing an additional neuron from the last convolutional layer of the network which is solely responsible for confidence estimation. This neuron has a sigmoid activation function to keep the confidence scores between 0 and 1. They teach the network to output high confidence scores, by weighting the softmax values of the network with the confidence score. The confidence is then penalized by a log-loss, which forces the predicted confidence scores to be far from 0 for in-distribution (training) samples. However, this approach requires extremely high regularization to work as intended.

Some metric learning based approaches also try to perform the task of OOD detection. For example, Masana et al. [18] propose a metric learning based method which uses a contrastive loss on a Siamese network to learn feature embeddings. However, their method requires samples from an OOD dataset to train.

There are also approaches which use an ensemble of classifiers [13] and Bayesian probabilistic models [15], [17] to tackle the problem of OOD detection. However, these methods are computationally expensive and have higher inference times compared to the previously mentioned approaches.

III. Methodology

To detect novel traffic signs we use OOD methods. More specifically, we consider threshold based detectors *Section III − A* and GAN based approaches *Section III − B*. Further, we compare these approaches with novelty detection methods i.e One-Class SVM in *Section III − C*

A. Threshold Based Detectors

Threshold-based detector methods do not require any additional pre-training of the network and can be used out-of-the-box on any pre-trained classifier to detect OOD samples. This can be used as a baseline to compare the performance of detectors on the OOD detection task. They feed each input sample $\boldsymbol{x}$ into the neural network, and calculate its softmax score $S(\boldsymbol{x})$ is calculated. Next, they compare this score to a threshold δ. The input $\boldsymbol{x}$ is considered to be in-distribution if the softmax score is above the threshold and is considered OOD, otherwise. This simple OOD detector g is formulated as,

$$g(\boldsymbol{x};\delta) = \begin{cases} 1 & \text{if} \max S(\boldsymbol{x}) \leq \delta, \\ 0 & \text{if} \max S(\boldsymbol{x}) > \delta. \end{cases} \tag{1}$$

Liang et al. [16] further improve the ability of the OOD detector in Eq. (1) by introducing temperature scaling and a pre-processing technique based on adding perturbations to the input data.

During training time, they do not apply temperature scaling. During test time, the temperature scaling modifies the standard softmax function with a temperature scaling parameter $T \in \mathbb{R}^{+}$ such that,

$$S_i(\boldsymbol{x};T) = \frac{\exp\left(f_i(\boldsymbol{x})/T\right)}{\sum_{j=1}^{N} \exp\left(f_j(\boldsymbol{x})/T\right)}, \tag{2}$$

where $\mathbf{x}$ denotes the input sample, $\boldsymbol{f} = (f_1, ..., f_N)$ represents the logits, i.e. the output of the last layer of the neural network classifying N classes and S denotes the modified softmax function. The network predicts an output label y using this temperature scaled softmax function. The authors show that during test time, a favorable selection of the scaling parameter T can help push the softmax scores of the in and out-of distribution samples further apart from each other, thus making the OOD samples easier to differentiate.

Liang et al. [16] also propose a pre-processing technique, which involves adding small controlled perturbations to the input data as follows:

$$\tilde{\boldsymbol{x}} = \boldsymbol{x} - \varepsilon \text{sign}(-\nabla_{\boldsymbol{x}} \log S_y(\boldsymbol{x};T)), \tag{3}$$

where ε is the perturbation magnitude. Adding controlled perturbations affects the in-distribution samples more than it does OOD samples, thus helping the classifier easily distinguish in-distribution from OOD samples. This enhanced detector is formulated similar to Eq. (1) as,

$$g(\boldsymbol{x};\delta,T,\varepsilon) = \begin{cases} 1 & \text{if} \max S(\tilde{\boldsymbol{x}};T) \leq \delta, \\ 0 & \text{if} \max S(\tilde{\boldsymbol{x}};T) > \delta. \end{cases}$$

B. Confidence Calibration using GAN

Lee et al. [14] propose a training technique to further improve the inherent property (as shown by Hendrycks and Gimpel [11]) of trained classifiers to assign higher softmax scores to in-distribution samples and lower softmax scores to OOD samples. They suggest that additionally minimizing the KL divergence loss between the softmax scores assigned by the classifier on OOD samples and the uniform distribution $\mathscr{U}(y)$, where y is the prediction of the network, should help the network learn to be less confident on OOD samples. To this end, they optimize

$$\begin{aligned} \min \; & E_{P_{\text{in}}(\widehat{\mathbf{x}},\widehat{y})}\left[-\log S(y=\widehat{y}|\widehat{\mathbf{x}}) \right] \\ & + \beta \cdot E_{P_{\text{out}}(\mathbf{x})}\left[KL(\mathscr{U}(y) \parallel S(y|\mathbf{x})) \right], \end{aligned} \tag{4}$$

where P_{in} represents the in-distribution, P_{out} represents the OOD and $\beta > 0$ is a penalty parameter. Thus, $P_{\text{out}}(\mathbf{x})$ refers to a sample taken from the out-distribution and $P_{\text{in}}(\widehat{\mathbf{x}},\widehat{y})$ refers to a sample and its corresponding ground truth taken from the in-distribution. The first term in the confidence loss corresponds to the standard label-based cross entropy loss used for the task of correctly classifying the categories of in-distribution samples. The second term of the confidence loss, i.e. the KL divergence term, forces the classifier to predict values closer to the uniform distribution for OOD samples. Therefore, the KL divergence term forces the classifier to be uncertain, i.e. it is unable to predict a high softmax value on OOD samples. Consequently, the classifier not only learns to perform well on its original classification task, but is also able to distinguish whether a given sample is from the OOD. However, to minimize the KL divergence term in Eq. (4), the authors need explicit samples from the OOD, to which they do not have access to during training. To tackle this problem the authors use a GAN to generate samples from the out-distribution. However, as the priori knowledge for OOD samples is not available, Lee et al. [14] propose to modify the original GAN loss so that it generates samples in the low-density region of the in-distribution. The original GAN loss [7] is formulated as,

$$\begin{aligned} \min_{G} \max_{D} \; & E_{P_{\text{in}}(\mathbf{x})}\left[\log D(\mathbf{x}) \right] \\ & + E_{P_{\text{pri}}(\mathbf{z})}\left[\log\left(1 - D(G(\mathbf{z}))\right) \right], \end{aligned} \tag{5}$$

where G is the Generator, D is the Discriminator and $P_{\text{pri}}(\mathbf{z})$ represents a latent variable $\mathbf{z}$ sampled from a prior distribution which is used by the generator. Optimizing this min-max objective forces $P_G \approx P_{\text{in}}$, i.e. forces the generator to generate samples from the in-distribution.

However, as the objective is to generate samples from P_{out}, the authors add an additional KL divergence loss term to Eq. (5) similar to the one in Eq. (4). This modified loss is

formulated as,

$$\min_G \max_D \underbrace{\beta \cdot E_{P_G(\mathbf{x})}\big[KL(\mathscr{U}(y) \parallel S(y|\mathbf{x}))\big]}_{\text{(a)}} + \underbrace{E_{P_{\text{in}}(\mathbf{x})}\big[\log D(\mathbf{x})\big] + E_{P_G(\mathbf{x})}\big[\log(1-D(\mathbf{x}))\big]}_{\text{(b)}}, \quad (6)$$

The first term of this loss is similar to the KL divergence loss term in Eq. (4). However, the generator distribution P_G replaces the P_{out} in this KL loss i.e.,the KL loss minimizes the samples generated by the generator, instead of needing explicit OOD samples. The second term corresponds to the original GAN loss in Eq. (5) which tries to generate in-distribution samples. However, term (a) in Eq. (6) forces the generator to generate samples further away from the in-distribution, as in-distribution samples would increase the KL divergence loss. On the contrary, if the generator generates samples which are too far away from the in-distribution boundary, the term (b) will be very high, i.e. the GAN loss forces the generator to create samples which are not too far away from the in-distribution boundary. The authors combine these 2 equations, i.e. Eq. (6) + Eq. (4), to jointly train the GAN and the classifier. Thus the KL divergence loss term in both equations not only encourages the GAN to generate samples in the low-density area of the in-distribution, but also forces the classifier to be uncertain on OOD samples. This joint confidence loss is formulated as follows:

$$\min_G \max_D \min \underbrace{E_{P_{\text{in}}(\widehat{\mathbf{x}},\widehat{y})}\big[-\log S(y=\widehat{y}|\widehat{\mathbf{x}})\big]}_{\text{(c)}} + \beta \cdot \underbrace{E_{P_G(\mathbf{x})}\big[KL(\mathscr{U}(y) \parallel S(y|\mathbf{x}))\big]}_{\text{(d)}} + \underbrace{E_{P_{\text{in}}(\widehat{\mathbf{x}})}\big[\log D(\widehat{\mathbf{x}})\big] + E_{P_G(\mathbf{x})}\big[\log(1-D(\mathbf{x}))\big]}_{\text{(e)}}, \quad (7)$$

where (c) + (d) correspond to the classification loss and (d) + (e) corresponds to the GAN loss. Thus, this joint confidence loss minimizes the KL divergence on low-density in-distribution samples, effectively helping the classifier to create a tighter bound on the in-distribution, without affecting the performance of the original classification task.

C. One-Class SVM

One-class SVMs [23] try to separate all data points from the origin with a hyperplane in a projected feature space. Further, they maximize the distance from the hyperplane to the origin. To this end they solve a quadratic optimization problem i.e.,

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\sum_{ij}\alpha_i\alpha_j k(\mathbf{x}_i,\mathbf{x}_j) \text{ subject to } 0 \le \alpha_i \le \frac{1}{\nu\ell}, \sum_i \alpha_i = 1 \quad (8)$$

where x_i denotes the hidden layer features of our CNN, $k(\cdot,\cdot)$ is an RBF kernel function, $\alpha_{i,j}$ are support vectors, ℓ is the number of training samples and ν is a hyperparameter.

IV. Experiments

In this section we compare the performance of OOD detectors on the task of detecting OOD traffic signs. In Section IV-A we give an overview of our in-distribution and OOD datasets. In Section IV-B we give a detailed overview of the pre-processing and data-augmentation techniques applied by us on the in-distribution dataset. In Section IV-C we explain our training setup and provide choices of the different hyperparameters that give us the best results. In Section IV-D we summarize the results of the state-of-the-art methods used by us on the task of novel traffic-sign detection.

For evaluation, we use the metrics proposed by Liang et al. [16]. Specifically, we compare the confidence scores from both in-distribution and OOD samples in the test set. From these scores, we compare the area under the ROC curve (AUROC), the FPR at 95% TPR and minimal detection error metrics. Further, we compute the area under the precision-recall curve, where we consider the in-distribution as positive class (AUPR In) and the OOD as the positive class AUPR Out.

A. Datasets

In the following, we describe the in-distribution and OOD datasets that we use for all our experiments.

1) In-Distribution Dataset: We choose the first 35 of 43 classes, of the GTSRB as our in-distribution. This dataset originally consists of 30,621 training images and 11,070 test images, each of size 32×32. After applying our extensive pre-processing and data-augmentation techniques, we extend the first 35 classes of the GTSRB (GTSRB-35) dataset to 612,420 images (see in Figure 2).

2) Out-of-Distribution Datasets: At test time, we consider the test-images from our training dataset (GTSRB-35) as in-distribution samples. For OOD samples we test our detectors with several image datasets, as listed below.

(1) The **GTSRB-last8** consists of the last 8 classes of the GTSRB, i.e the classes of GTSRB which were not used during the training phase. This dataset comprises of 4,440 images, each of size 32×32. We show a test sample from each of the 8 classes in Figure 3.
(2) The **Private Dataset** consists of 3 classes, i.e. Bike and Pedestrians sign, No Stopping Zone sign and Unknown

Fig. 2: Samples of the 35 classes used for training our network.

Fig. 3: Samples from the last 8 classes of GTSRB, which are used by us for evaluation.

traffic-signs. The Unknown sign class consists of several traffic signs which are region specific and not too crucial for everyday usage. We obtain this dataset from a drive in an urban setting. This private dataset comprises of 1,293 images, each image has been resized to a size of 32×32. We also show some sample images from each of the 3 classes in Figure 4.

B. Pre-Processing

We apply some specialized pre-processing and data augmentation techniques to increase the size of the input data from GTSRB to improve robustness of our trained classifiers. We cannot use standard augmentation techniques such as random flipping and rotation as they can change the meaning of the sign.
To overcome this problem, we use different augmentations depending on the class. We always apply small geometric distortions (e.g., rotation, projective transforms). We flip signs horizontally or vertically only if they are symmetric (i.e., the class label does not change), or if the transformation yields an image with a different class label. For the latter images, we change the corresponding label after this transformation.(e.g., turn left - turn right)

Fig. 4: Samples from our private dataset, which are used for evaluation. The first sign is from the Bike & Pedestrians class, the second from the No Stopping Zone class and the third a random sign from the Unknown class.

C. Training Setup

We perform all our experiments using a simple VGG-13 network [24]. We adopt the same CNN architecture and hyperparameters for our VGG-13 network as Lee et al. [14]. Our network is able to achieve a classification test accuracy of 98% on the first 35 classes of the GTSRB dataset (GTSRB-35).

D. Results

In this section, we show the results of our baseline approaches in Section IV-D.1, threshold based approaches (ODIN) in Section IV-D.2 and GAN-based approaches in Section IV-D.3. We compare the results of all our experiments with that of the Baseline detector [11], so as to get a qualitative understanding of the performance of each approach. We also compare the simple softmax based thresholding detector with novelty detection methods like the One-Class SVM [23] and also with a supervised method i.e. the linear SVM, so as to get an upper bound on the performance of the OOD methods.

1) Baseline Methods: In this section, we explain our three baseline approaches, i.e. simple softmax thresholding, one-class SVM and supervised linear SVM. For these approaches we first train a standard CNN classifier. The simple thresholding approach detects OOD images based on the maximum softmax scores.
For the one-class SVM we use an RBF kernel and sample 10,000 images from the training set. We extract 512 dimensional features from the last hidden layer of our CNN, which we use for training. For the one-class SVM we apply a grid search on the parameters nu and gamma, and find that setting ν to 0.0001 and γ to 3.2 and 2.3 works best for GTSRB-last8 and our private dataset, respectively.
For the supervised linear SVM baseline, we further use

In distribution dataset	Out-of-distribution dataset	Method	FPR (95% TPR) ↓	Detection Error ↓	AUROC ↑	AUPR In ↑	AUPR Out ↑
GTSRB-35	GTSRB-last8	**Baseline [11]**	100.0	37.4	26.5	57.3	94.7
		ODIN	28.1	10.4	95.1	96.2	92.6
		GAN-based approach	24.1	10.5	91.7	92.1	94.2
		One-class SVM	**12.1**	**5.7**	**97.2**	**98.1**	**95.1**
		Linear SVM*	0.1	1.7	99.7	99.7	99.5
	Private Dataset	**Baseline [11]**	100.0	8.7	83.7	85.9	98.9
		ODIN	1.0	2.7	99.4	99.5	99.2
		GAN-based approach	**0.7**	**2.5**	**99.5**	**99.6**	**99.4**
		One-class SVM	1.8	2.7	98.8	98.0	98.8
		Linear SVM*	0.8	1.7	99.3	99.0	99.5

TABLE I: Distinguishing in- and OOD test set data for traffic-sign classification. All values are percentages. ↑ indicates larger value is better, and ↓ indicates lower value is better.
* denotes supervised learning

samples from the train split of the GTSRB-last8 and our private dataset (which are not used during evaluation) for training. We also perform a grid search on the penalty parameter C and set it to 1 and 10 for GTSRB-last8 and our private dataset, respectively.

2) Threshold based Methods: In this section, we summarize the results of the ODIN detector proposed by Liang et al. [16] in Table I. We empirically set the temperature parameter to 1000 and the perturbation magnitude to 0.0034 for both our target datasets.

Fig. 5: Images generated by the GAN in the low-density region of the in-distribution

3) Confidence Calibration using GAN: In this section we summarize the results of the GAN based OOD method proposed by Lee et al. [14] in Table I. We choose the penalty parameter for the KL loss, i.e. β to be empirically 1.3.
We also show some of the traffic-sign images generated by the GAN to verify that the generated samples are indeed in the low-density region of the in-distribution (GTSRB-35) in Figure 5. Interestingly, as we observe from the first and second images generated by the GAN, the generated samples are hard to recognize even by humans. For example, the first image could be either interpreted as 50km/h or 60km/h sign. Similarly, the second image could be interpreted as either No Vehicles (Circular sign with red boundary) or Yield (Triangular Sign with red boundary). Thus, this suggests that the generator does indeed generate samples which look similar to the in-distribution. However, as these samples are extremely difficult to recognize even by humans, we can conclude that the generated samples are indeed from the low-density region of the in-distribution.

To summarize our results, we find that the baseline one-class SVM yields the best results on the GTSRB-last8 dataset and the GAN-based approach performs best on our private dataset. Interestingly, we find that the supervised linear SVM serves as a reasonable estimate of the upper-bound on the task of OOD traffic sign detection.

V. Conclusion

In this paper, we proposed a benchmark for the task of novel traffic sign detection by using current state-of-the-art OOD detection and novelty detection methods. We showed that the one-class SVM performs best on the challenging task of detecting the GTSRB-last8 dataset and the GAN-based approach performs best on detecting our private dataset as OOD. For future work, we intend to use this benchmark to help autonomous vehicles learn new traffic sign classes incrementally. We also plan to experiment with metric learning based approaches and a variety of divergence losses to further improve performance of OOD detectors.

VI. Acknowledgements

This work was supported by the Austrian Research Promotion Agency (FFG) under the project *DGT - Dynamic Ground Truth (860820)*. We also thank *Joanneum Research Digital* for providing us with anonymized imagery, which made our dataset evaluations possible.

References

[1] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad, "Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission," in *Proceedings of ACM SIGKDD*. ACM, 2015.

[2] H. S. Christopher D Manning, "Foundations of statistical natural language processing," *MIT Press*, vol. 999, 1999.

[3] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, 1995.

[4] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves." in *ICML*, 2006.

[5] T. DeVries and G. W. Taylor, "Learning confidence for out-of-distribution detection in neural networks," *arXiv preprint arXiv:1802.04865*, 2018.

[6] T. Fawcett, "An introduction to roc analysis," *Pattern recognition letters*, 2006.

[7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NIPS*, 2014.

[8] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples. corr (2015)."

[9] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," *arXiv preprint arXiv:1706.04599*, 2017.

[10] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, *et al.*, "Deep speech: Scaling up end-to-end speech recognition," *arXiv preprint arXiv:1412.5567*, 2014.

[11] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," in *ICLR*, 2017.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.

[13] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *NIPS*, 2017.

[14] K. Lee, H. Lee, K. Lee, and J. Shin, "Training confidence-calibrated classifiers for detecting out-of-distribution samples," *arXiv preprint arXiv:1711.09325*, 2017.

[15] Y. Li and Y. Gal, "Dropout inference in Bayesian neural networks with alpha-divergences," in *ICML*, 2017.

[16] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," *ICLR*, 2018.

[17] C. Louizos and M. Welling, "Multiplicative normalizing flows for variational Bayesian neural networks," in *ICML*, 2017.

[18] M. Masana, I. Ruiz, J. Serrat, J. van de Weijer, and A. M. Lopez, "Metric learning for novelty and anomaly detection," *BMVC*, 2018.

[19] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013.

[20] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[21] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[22] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets," *PloS one*, vol. 10, no. 3, 2015.

[23] B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, and J. C. Platt, "Support vector method for novelty detection," in *NIPS*, 2000.

[24] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.

[25] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The German Traffic Sign Recognition Benchmark: A multi-class classification competition," in *IEEE International Joint Conference on Neural Networks*, 2011.

[26] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *International Conference on Learning Representations*, 2014. [Online]. Available: http://arxiv.org/abs/1312.6199